

PTS Project 实验报告 (c++实现)

刘心悦
161220085

一、 实验要求

本项目的总体目标是要实现一个标准（确定性、双向无限纸带）图灵机的模拟器，并且能够在上面正确模拟运行符合特定语法的图灵机程序。

二、 设计思路

项目需要实现一个标准图灵机程序解析器，能够以符合语法描述的图灵机程序作为解析器的输入，模拟一个标准的图灵机。所以需要在程序中实现一个 TM 类，类中的私有成员储存图灵机的状态集 Q ，输入符号集 Σ ，纸带符号集 Γ ，初始状态 q_0 ，空格符号 B ，终结状态集 F 和转移函数 δ 。同时类中还要有若干成员函数负责输入图灵机、输入纸带、运行图灵机、打印图灵机输出等任务。

三、 具体实现

项目代码在 src 文件夹下，由 main.cpp, tm.h, tm.cpp 三个文件构成。tm.h 中构造了 TM 类，tm.cpp 中是 TM 类中成员函数的实现。main.cpp 调用 tm.h，读入图灵机和纸带并运行，最后打印输出结果。

TM 中的数据成员如下图所示，集合均用 vector 实现。其中瞬时描述 (ID) 用于记录运行过程中当前 TM 的状态。

```
vector<string> Q;           //状态集
vector<char> S;             //输入符号集
vector<char> T;             //纸带符号集
string q0;                  //初始状态
char B;                     //空格符号
vector<string> F;           //终结状态集
vector<transi> delta;       //转移函数集
//瞬时描述
struct {
    string tape;
    int step;
    int head;
    string state;
} ID;
```

TM 中的成员函数 **readTM** 用于从 xxx.tm 文件中读入图灵机。readTM 会一行一行地处理文件中的内容，首先它会删去该行中分号后面的内容（注释），如果分号在开头则舍弃改行。然后识别剩余字符串开头的内容，交给不同的行读入函数处理。例如字符串以 "#Q =" 开头，则把该字符串交给成员函数 readQ 处理。

TM 中的成员函数 **printID** 用于打印图灵机当前的瞬时描述 (ID) 到 console.txt。

TM 中的成员函数 **run** 用于在当前图灵机上运行纸带上的内容。run 函数首先会检查纸带上的字符是否都属于纸带字符集，如果不是则在 result.txt 中输出 Error，运行结

束；如果是，则填好 TM 的瞬时描述，寻找匹配的转移函数，改变瞬时描述。不断重复上述过程，直到抵达终结状态，或者未找到匹配的转移函数，运行结束，并在 result.txt 中打印最终纸带上的内容。

main 函数中生成了一个 TM 实例，调用 readTM 从 xxx.tm 中读入图灵机结构，然后依次从 input.txt 中读入纸带内容，并分别调用 run 函数运行。

四、 编译运行说明

在根目录下控制台输入 **make** 指令即可得到可执行文件 turing。再输入 **./turing case-directory** 即可在当前 case-directory 目录下运行图灵机程序，并得到 result.txt 和 console.txt 两个输出文件。输入 **make clean** 指令即可删除 turing 和所有文件夹中生成的输出文件

五、 演示

示例 Input.txt 如下所示：

```
0x1=11
1x=1
11x111=111111
1x111=111
```

执行 **./turing case1** 后 result.txt 中结果如下：

```
Error
False
True
True
```

符合预期。

六、 写在最后

本次实验的 TM 的实现原理非常简单，只是在对于 xxx.tm 文件的输入处理时花了一些功夫，因为需要处理注释，一开始没有想到好的处理方法。之后采用了一行一行地读入，删去该行注释部分再进行处理的方式，较好地解决了这个问题。

在实现 Assignment 3 - Problem 2 中两个图灵机对应的图灵机程序时也稍费了些周章，因为一开始不太熟练，所以用了很多时间才勉强构造出了第一个图灵机程序。不过熟练之后就没什么问题了，十几分钟就完成了第二个图灵机程序的构造，还是熟能生巧吧。